

WEST Search History

[Hide Items](#)[Restore](#)[Clear](#)[Cancel](#)

DATE: Thursday, November 30, 2006

Hide?	<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=NO; OP=OR</i>	
<input type="checkbox"/>	L6	l4 and L5	1
<input type="checkbox"/>	L5	(l1 or l2 or l3) and ((client\$ or consumer\$ or user\$ or customer\$) near fail\$)	4
<input type="checkbox"/>	L4	(l1 or l2 or l3) and (replica\$ with (database\$ or (data adj1 base\$) or dbms) with (file or files or folder or folders) with (timestamp\$ or (time adj1 stamp\$) or time-stamp\$))	1
<input type="checkbox"/>	L3	romine-william-j.in.	2
<input type="checkbox"/>	L2	kalderon-eyal.in.	4
<input type="checkbox"/>	L1	aronoff-eyal.in.	4

END OF SEARCH HISTORY

10/661,412

WEST Search History

DATE: Thursday, November 30, 2006

Hide?	<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>
		<i>DB=PGPB,USPT,USOC; PLUR=NO; OP=OR</i>	
<input type="checkbox"/>	L46	L45 and (second adj1 (database or (data adj1 base) or dbms))	4
<input type="checkbox"/>	L45	L44 and (first adj1 (database or (data adj1 base) or dbms))	4
<input type="checkbox"/>	L44	L43 and (timestamp\$ or (time adj1 stamp\$) or time-stamp\$)	20
<input type="checkbox"/>	L43	L42 and (replicat\$ near (file or files or folder or folders))	50
<input type="checkbox"/>	L42	(136 or 137 or 138 or 139 or 140) and (replicat\$ near (database\$ or (data adj1 base\$)))	433
<input type="checkbox"/>	L41	(136 or 137 or 138 or 139 or 140) and (replicat\$ with (file or files or folder or folders) with (timestamp\$ or (time adj1 stamp\$) or time-stamp\$) with (request\$ or quer\$ or search\$ or inquir\$ or enquir\$ or question\$) with (database\$ or (data adj1 base\$)))	0
<input type="checkbox"/>	L40	709/203.ccls.	7088
<input type="checkbox"/>	L39	7070/206.ccls.	0
<input type="checkbox"/>	L38	707/200-201.ccls.	3938
<input type="checkbox"/>	L37	707/8.ccls.	1111
<input type="checkbox"/>	L36	707/1-5.ccls.	15909
		<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=NO; OP=OR</i>	
<input type="checkbox"/>	L35	L34 and (client\$ or consumer\$ or user\$ or customer\$)	72
<input type="checkbox"/>	L34	L33 and (timestamp\$ or (time adj1 stamp\$) or time-stamp\$)	73
<input type="checkbox"/>	L33	L32 and (replicat\$ near (database\$ or (data adj1 base\$)))	159
<input type="checkbox"/>	L32	L31 and (second adj1 (database or (data adj1 base) or dbms))	4871
<input type="checkbox"/>	L31	(first adj1 (database or (data adj1 base) or dbms))	6360
<input type="checkbox"/>	L30	L29 and (timestamp\$ or (time adj1 stamp\$) or time-stamp\$)	76
<input type="checkbox"/>	L29	L27 and (replicat\$ near (file or files or folder or folders))	136
<input type="checkbox"/>	L28	(replicat\$ with (file or files or folder or folders) with (timestamp\$ or (time adj1 stamp\$) or time-stamp\$) with (request\$ or quer\$ or search\$ or inquir\$ or enquir\$ or question\$) with (database\$ or (data adj1 base\$)))	1
<input type="checkbox"/>	L27	(replicat\$ near (database\$ or (data adj1 base\$)))	1728
<input type="checkbox"/>	L26	L24 and L25	1
<input type="checkbox"/>	L25	L23 and (replicat\$ near (timestamp\$ or (time adj1 stamp\$) or time-stamp\$))	2
<input type="checkbox"/>	L24	L23 and ((search\$ or request\$ or quer\$ or question\$ or ask or asks or asking or asked or inquir\$ or enquir\$) with (access or accessing) with (file or files or folder or folders))	19

10/661,412

<input type="checkbox"/>	L23 (L21 or L22) and (replicat\$ near (database\$ or (data adj1 base\$)))	577
<input type="checkbox"/>	L22 replicat\$.ab.	23027
<input type="checkbox"/>	L21 replicat\$.ti. <i>DB=USPT; PLUR=NO; OP=OR</i>	6033
<input type="checkbox"/>	L20 (L18 or L19) and timestamp\$	22
<input type="checkbox"/>	L19 L17 and (replicat\$ with (file or files or folder or folders))	107
<input type="checkbox"/>	L18 L17 and (replicat\$ near (file or files or folder or folders))	53
<input type="checkbox"/>	L17 replicat\$.ti.	929
<input type="checkbox"/>	L16 L14 and timestamp\$	1
<input type="checkbox"/>	L15 L1 and (replicat\$ near (file or files or folder or folders))	1
<input type="checkbox"/>	L14 L1 and (replicat\$ with (file or files or folder or folders))	11
<input type="checkbox"/>	L13 (replicat\$ near timestamp)	2
<input type="checkbox"/>	L12 L1 and (replicat\$ near timestamp)	0
<input type="checkbox"/>	L11 L1 and (replicat\$ with timestamp)	4
<input type="checkbox"/>	L10 (replicat\$ with timestamp) <i>DB=PGPB; PLUR=NO; OP=OR</i>	29
<input type="checkbox"/>	L9 L1 and (replicat\$ with timestamp)	0
<input type="checkbox"/>	L8 L1 and (replicat\$ near timestamp)	0
<input type="checkbox"/>	L7 (replicat\$ with timestamp with (file or files or folder or folder\$)) <i>DB=USPT; PLUR=NO; OP=OR</i>	11
<input type="checkbox"/>	L6 L1 and (timestamp with (folder or folders or file or files))	1
<input type="checkbox"/>	L5 L3 and (timestamp with (folder or folders or file or files))	0
<input type="checkbox"/>	L4 L3 and (replicat\$ near timestamp near (file or files or folder or folder\$))	0
<input type="checkbox"/>	L3 L1 and databases	200
<input type="checkbox"/>	L2 L1 and (replicat\$ with timestamp with (file or files or folder or folder\$))	0

(5596745 5423022 5721904 5909584 6031978 6092063 6247055 6314430
6446092 6941310 6058393 5303367 5555409 5333251 5471611 5504885
5757669 5768587 5953719 6182117 6219670 6219670 6557012 6584476
6598059 6631386 6813623 5835904 5388257 5966706 5561763 5640555
5694598 5717919 5737536 5768577 5781739 5835755 5884324 5930793
6076092 6125401 6151602 6226651 6226651 6243710 5493728 5561793
5615337 5632015).pn. (5632031 5642504 5644763 5659727 5664181 5668987
5680618 5682524 5687365 5713018 5721895 5732262 5742806 5778223
5794229 5794228 5799310 5809497 5812996 5812983 5819277 5822749
5826253 5832484 5834856 5842196 5873086 5872969 5884327 5893912
5907847 5918059 5918191 5918225 5937415 5937401 5950188 5953715
5951661 5960420 5970494 5987454 5991776 6006224 6012054 6014674
6023269 6038633 6044216 6047291).pn. (6052685 6052281 6055647 6058394
6078925 6122627 6134540 6138120 6173290 6181336 6192370 6199062
6219675 6219790 6226637 6226659 6219675 6219790 6226637 6226659
6240417 6256626 6278992 6285997 6289334 6289343 6341279 6353826
6356920 6356946 6363387 6430556 6438559 6442541 6470343 6470360

6473504 6477527 6477540 6487551 6502088 6505199 6507834 6564215
6571232 6574639 6578043 6591266 6606626 6609123).pn. (6609126 6691166
6708186 6714928 6728710 6732084 6735582 6738763 6738866 6769048
6772159 6792432 6816864 6862620 6874010 6925515 6938041 6938043
5357630 5644718 6032189 5850517 5899990 6157941 6199111 6209029
6226750 6226750 5515510 5539621 5596748 5706499 5832191 5862327
5870765 5870759 5884325 5887132 5926816 5943671 5987465 5999941
6088728 6098099 6115744 6154743 6163878 6178418 6178439 6199110).pn.
(6216151 6226684 6216151 6226684 6243751 6247017 5301280 5450123
5550982 5675786 5689697 5732219 5742675 5745754 5754772 5793966
L1 5835856 5857201 5872973 5875234 5893096 5905868 5938737 5951636 288
5956697 5974441 5983268 5999947 6012098 6012100 6016496 6018745
6023698 6041309 6058418 6061727 6065051 6064666 6073172 6081601
6088515 6092099 6094531 6098108 6098093 6100891 6107961 6134533
6141325 6157630).pn. (6157927 6192390 6195329 6202023 6202085 6219712
6223219 6223220 6219712 6223219 6223220 6230194 6233565 6237041
6240186 6240441 6246678 6249291 6272332 6272675 6282281 6292555
6292830 6295502 6304576 6304881 6313835 6317544 6320974 6324542
6330562 6330566 6353448 6356912 6362836 6381627 6389460 6418214
6421787 6434141 6442598 6446109 6446219 6456308 6466570 6473786
6509828 6549538 6564258 6631512).pn.

END OF SEARCH HISTORY


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

relocation with timestamps with databases with file and client a



THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used

relocation with timestamps with databases with file and client and failure

Found 27,836 of 192,876

Sort results by

relevance


[Save results to a Binder](#)
[Try an Advanced Search](#)

Display results

expanded form


[Search Tips](#)
[Try this search in The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Deciding when to forget in the Elephant file system](#)



Douglas S. Santry, Michael J. Feeley, Norman C. Hutchinson, Alistair C. Veitch, Ross W. Carton, Jacob Ofir

 December 1999 **ACM SIGOPS Operating Systems Review , Proceedings of the seventeenth ACM symposium on Operating systems principles SOSP '99**, Volume 33 Issue 5

Publisher: ACM Press

 Full text available: [pdf \(1.61 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Modern file systems associate the deletion of a file with the immediate release of storage, and file writes with the irrevocable change of file contents. We argue that this behavior is a relic of the past, when disk storage was a scarce resource. Today, large cheap disks make it possible for the file system to protect valuable data from accidental delete or overwrite. This paper describes the design, implementation, and performance of the Elephant file system, which automatically retains all impo ...

2 [Industrial sessions: beyond relational tables: Coordinating backup/recovery and data consistency between database and file systems](#)



Suparna Bhattacharya, C. Mohan, Karen W. Brannon, Inderpal Narang, Hui-I Hsiao, Mahadevan Subramanian

 June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data SIGMOD '02**

Publisher: ACM Press

 Full text available: [pdf \(1.44 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Managing a combined store consisting of database data and file data in a robust and consistent manner is a challenge for database systems and content management systems. In such a hybrid system, images, videos, engineering drawings, etc. are stored as files on a file server while meta-data referencing/indexing such files is created and stored in a relational database to take advantage of efficient search. In this paper we describe solutions for two potentially problematic aspects of such a data ...

Keywords: DB2, content management, database backup, database recovery, datalinks

3

[File servers for network-based distributed systems](#)

10/661,412

 Liba Svobodova
December 1984 **ACM Computing Surveys (CSUR)**, Volume 16 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(4.23 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

4 Transaction-centric reconciliation in disconnected client-server databases

October 2004 **Mobile Networks and Applications**, Volume 9 Issue 5

Publisher: Kluwer Academic Publishers

Full text available:  [pdf\(205.54 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

As mobile computing devices become more and more common, mobile databases are becoming popular. An important feature of these database systems is their ability to allow optimistic replication of data by providing disconnected mobile devices the ability to perform local updates. The key problem to this approach is the reconciliation problem, i.e. the problem of serializing potentially conflicting updates from disconnected clients on all replicas of the database. Reconciliation of conflicting u ...

Keywords: databases, disconnected operation, reconciliation, serializability testing

5 ARIES/CSA: a method for database recovery in client-server architectures

 C. Mohan, Inderpal Narang
May 1994 **ACM SIGMOD Record , Proceedings of the 1994 ACM SIGMOD international conference on Management of data SIGMOD '94**, Volume 23 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(1.33 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents an algorithm, called ARIES/CSA (Algorithm for Recovery and Isolation Exploiting Semantics for Client-Server Architectures), for performing recovery correctly in client-server (CS) architectures. In CS, the server manages the disk version of the database. The clients, after obtaining database pages from the server, cache them in their buffer pools. Clients perform their updates on the cached pages and produce log records. The log records are buffered loca ...

6 BASE: Using abstraction to improve fault tolerance

 Miguel Castro, Rodrigo Rodrigues, Barbara Liskov
August 2003 **ACM Transactions on Computer Systems (TOCS)**, Volume 21 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(438.18 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software errors are a major cause of outages and they are increasingly exploited in malicious attacks. Byzantine fault tolerance allows replicated systems to mask some software errors but it is expensive to deploy. This paper describes a replication technique, BASE, which uses abstraction to reduce the cost of Byzantine fault tolerance and to improve its ability to mask software errors. BASE reduces cost because it enables reuse of off-the-shelf service implementations. It improves availability ...

Keywords: Byzantine fault tolerance, N-version programming, asynchronous systems, proactive recovery, state machine replication

7 Persistent memory: a storage architecture for object-oriented database systems

Satish M. Thatte

September 1986 **Proceedings on the 1986 international workshop on Object-oriented database systems**

Publisher: IEEE Computer Society Press

Full text available:  [pdf\(1.13 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Object-oriented databases are needed to support database objects with a wide variety of types and structures. A persistent memory system provides a storage architecture for long-term, reliable retention of objects with rich types and structures in the virtual memory itself. It is based on a uniform memory abstraction, which eliminates the distinction between transient objects (data structures) and persistent objects (files and databases), and therefore, allows the same set of powerful and f ...

8 **Reliable commit and optimistic concurrency control for dynamically reconfigurable distributed databases**

Sue M. Zajac, Fred J. Maryanski

December 1986 **Proceedings of the 1986 ACM SIGSMALL/PC symposium on Small systems**

Publisher: ACM Press

Full text available:  [pdf\(720.47 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper addresses the problem of maintaining high availability in a dynamically reconfigurable distributed database system. A dynamically reconfigurable distributed database system is one in which clients and/or servers may attach to or detach from the system at will. Reliable transaction management and commit protocols which handle both these absences and unexpected failures are presented.

9 **BASE: using abstraction to improve fault tolerance**

Rodrigo Rodrigues, Miguel Castro, Barbara Liskov

October 2001 **ACM SIGOPS Operating Systems Review , Proceedings of the eighteenth ACM symposium on Operating systems principles SOSP '01**, Volume 35 Issue 5

Publisher: ACM Press

Full text available:  [pdf\(1.47 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Software errors are a major cause of outages and they are increasingly exploited in malicious attacks. Byzantine fault tolerance allows replicated systems to mask some software errors but it is expensive to deploy. This paper describes a replication technique, BASE, which uses abstraction to reduce the cost of Byzantine fault tolerance and to improve its ability to mask software errors. BASE reduces cost because it enables reuse of off-the-shelf service implementations. It improves availability ...

10 **Client-server computing in mobile environments**

Jin Jing, Abdelsalam Sumi Helal, Ahmed Elmagarmid

June 1999 **ACM Computing Surveys (CSUR)**, Volume 31 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(233.31 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Recent advances in wireless data networking and portable information appliances have engendered a new paradigm of computing, called mobile computing, in which users carrying portable devices have access to data and information services regardless of their physical location or movement behavior. In the meantime, research addressing information access in mobile environments has proliferated. In this survey, we provide a concrete framework and categorization of the various way ...

Keywords: application adaptation, cache invalidation, caching, client/server, data dissemination, disconnected operation, mobile applications, mobile client/server, mobile computing, mobile data, mobility awareness, survey, system application

11 FAB: building distributed enterprise disk arrays from commodity components



Yasushi Saito, Svend Frølund, Alistair Veitch, Arif Merchant, Susan Spence

October 2004 **ACM SIGARCH Computer Architecture News**, **ACM SIGOPS Operating Systems Review**, **ACM SIGPLAN Notices**, **Proceedings of the 11th international conference on Architectural support for programming languages and operating systems ASPLOS-XI**, Volume 32, 38, 39 Issue 5, 5, 11

Publisher: ACM Press

Full text available:  [pdf\(671.67 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes the design, implementation, and evaluation of a Federated Array of Bricks (FAB), a distributed disk array that provides the reliability of traditional enterprise arrays with lower cost and better scalability. FAB is built from a collection of *bricks*, small storage appliances containing commodity disks, CPU, NVRAM, and network interface cards. FAB deploys a new majority-voting-based algorithm to replicate or erasure-code logical blocks across bricks and a reconfigurati ...

Keywords: consensus, disk array, erasure coding, replication, storage, voting

12 Decentralized storage systems: Taming aggressive replication in the Pangaea wide-area file system



Yasushi Saito, Christos Karamanolis, Magnus Karlsson, Mallik Mahalingam

December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue S1

Publisher: ACM Press

Full text available:  [pdf\(1.93 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Pangaea is a wide-area file system that supports data sharing among a community of widely distributed users. It is built on a symmetrically decentralized infrastructure that consists of commodity computers provided by the end users. Computers act autonomously to serve data to their local users. When possible, they exchange data with nearby peers to improve the system's overall performance, availability, and network economy. This approach is realized by aggressively creating a replica of a file w ...

13 Distributed file systems: concepts and examples



Eliezer Levy, Abraham Silberschatz

December 1990 **ACM Computing Surveys (CSUR)**, Volume 22 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(5.33 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The purpose of a distributed file system (DFS) is to allow users of physically distributed computers to share data and storage resources by using a common file system. A typical configuration for a DFS is a collection of workstations and mainframes connected by a local area network (LAN). A DFS is implemented as part of the operating system of each of the connected computers. This paper establishes a viewpoint that emphasizes the dispersed structure and decentralization of both data and con ...

14 Paradigms for process interaction in distributed programs



Gregory R. Andrews

March 1991 **ACM Computing Surveys (CSUR)**, Volume 23 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(3.77 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Distributed computations are concurrent programs in which processes communicate by message passing. Such programs typically execute on network architectures such as networks of workstations or distributed memory parallel machines (i.e., multicomputers such as hypercubes). Several paradigms—examples or models—for process interaction in distributed computations are described. These include networks of filters, clients, and servers, heartbeat algorithms, probe/echo algorithms, broa ...


Keywords: clients and servers, distributed and parallel algorithms, distributed programming, distributed programming methods, heartbeat algorithms, networks of filters, patterns for interprocess communication, probe/echo algorithms, replicated servers, token-passing algorithms

15 Log files: an extended file service exploiting write-once storage



R. Finlayson, D. Cheriton

November 1987 **ACM SIGOPS Operating Systems Review , Proceedings of the eleventh ACM Symposium on Operating systems principles SOSP '87**, Volume 21
Issue 5

Publisher: ACM PressFull text available:  [pdf\(1.07 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


A log service provides efficient storage and retrieval of data that is written sequentially (append-only) and not subsequently modified. Application programs and subsystems use log services for recovery, to record security audit trails, and for performance monitoring. Ideally, a log service should accommodate very large, long-lived logs, and provide efficient retrieval and low space overhead. In this paper, we describe the design and implementation of the Clio log service. Clio pr ...

16 Cheap recovery: a key to self-managing state



Andrew C. Huang, Armando Fox

February 2005 **ACM Transactions on Storage (TOS)**, Volume 1 Issue 1

Publisher: ACM PressFull text available:  [pdf\(1.24 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Cluster hash tables (CHTs) are key components of many large-scale Internet services due to their highly-scalable performance and the prevalence of the type of data they store. Another advantage of CHTs is that they can be designed to be as self-managing as a cluster of stateless servers. One key to achieving this extreme manageability is reboot-based recovery that is predictably fast and has modest impact on system performance and availability. This "cheap" recovery mechanism simplifies manage ...

Keywords: Cluster hash table, manageability, quorum replication, storage systems design

17 A quorum-consensus replication method for abstract data types



Maurice Herlihy

February 1986 **ACM Transactions on Computer Systems (TOCS)**, Volume 4 Issue 1

Publisher: ACM PressFull text available:  [pdf\(1.66 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Replication can enhance the availability of data in distributed systems. This paper introduces a new method for managing replicated data. Unlike many methods that


support replication only for uninterpreted files, this method systematically exploits type-specific properties of objects such as sets, queues, or directories to provide more effective replication. Each operation requires the cooperation of a certain number of sites for its successful completion. A quorum for an operation is any s ...

18 Replication in the harp file system



Barbara Liskov, Sanjay Ghemawat, Robert Gruber, Paul Johnson, Liuba Shriru
September 1991 **ACM SIGOPS Operating Systems Review , Proceedings of the thirteenth ACM symposium on Operating systems principles SOSP '91**, Volume 25 Issue 5

Publisher: ACM Press

Full text available:  [pdf\(1.60 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


This paper describes the design and implementation of the Harp file system. Harp is a replicated Unix file system accessible via the VFS interface. It provides highly available and reliable storage for files and guarantees that file operations are executed atomically in spite of concurrency and failures. It uses a novel variation of the primary copy replication technique that provides good performance because it allows us to trade disk accesses for network communication. Harp is intended to be u ...

19 Manageability, availability, and performance in porcupine: a highly scalable, cluster-based mail service



Yasushi Saito, Brian N. Bershad, Henry M. Levy
August 2000 **ACM Transactions on Computer Systems (TOCS)**, Volume 18 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(2.52 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes the motivation, design and performance of Porcupine, a scalable mail server. The goal of Porcupine is to provide a highly available and scalable electronic mail service using a large cluster of commodity PCs. We designed Porcupine to be easy to manage by emphasizing dynamic load balancing, automatic configuration, and graceful degradation in the presence of failures. Key to the system's manageability, availability, and performance is that sessions, data, and underlying ...

Keywords: cluster, distributed systems, email, group membership protocol, load balancing, replication

20 Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining



Robbert Van Renesse, Kenneth P. Birman, Werner Vogels
May 2003 **ACM Transactions on Computer Systems (TOCS)**, Volume 21 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(341.62 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Scalable management and self-organizational capabilities are emerging as central requirements for a generation of large-scale, highly dynamic, distributed applications. We have developed an entirely new distributed information management system called Astrolabe. Astrolabe collects large-scale system state, permitting rapid updates and providing on-the-fly attribute aggregation. This latter capability permits an application to locate a resource, and also offers a scalable way to track sys ...





Keywords: Aggregation, epidemic protocols, failure detection, gossip, membership, publish-subscribe, scalability

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



replicate with databases with timestamps with

Search

[Advanced Scholar Search](#)
[Scholar Preferences](#)
[Scholar Help](#)

The following words are very common and were not included in your search: with with with with with. [\[details\]](#)

Scholar [All articles](#) [Recent articles](#) Results 1 - 10 of about 1,710 for **replicate with databases with timest**

All Results

[M Satyanarayan...](#)

[B Liskov](#)

[J Howard](#)

[M Theimer](#)

[D Barabará](#)

[Availability and consistency tradeoffs in the Echo distributed filesystem - group of 2 »](#)

A Hisgen, A Birrell, T Mann, M Schroeder, G Swart - Workstation Operating Systems, 1989., Proceedings of the ..., 1989 - [ieeexplore.ieee.org](#)

... conflicting updates are resolved using **timestamps**: the most ... the global name service

replication scheme is ... ple and efficient implementation for small **databases**. ...

[Cited by 37](#) - [Related Articles](#) - [Web Search](#)

[File and Object Replication in Data Grids - group of 41 »](#)

H Stockinger, A Samar, K Holtman, B Allcock, I ... - Cluster Computing, 2002 - Springer

... such as **file** size and modify **time-stamps**) and the ... LDAP protocol to interface with the **database** backend. We do not currently distribute or **replicate** the replica ...

[Cited by 126](#) - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

[Replication in the harp file system - group of 20 »](#)

B Liskov, S Ghemawat, R Gruber, P Johnson, L ... - Proceedings of the thirteenth ACM symposium on Operating ..., 1991 - [portal.acm.org](#)

... The systems use a multipart **timestamp** scheme ... incon- sistencies are detected using

the muk.ipart **timestamps**.) In contrast, Harp uses **replication** only to provide ...

[Cited by 167](#) - [Related Articles](#) - [Web Search](#) - [Library Search](#)

[Gold Rush: Mobile Transaction Middleware with Java-Object Replication - group of 5 »](#)

MA Butrico, H Chang, A Cocchi, NH Cohen, DG Shea, ... - [usenix.org](#)

... If the **timestamps** are different, we transmit both the ... Unlike systems that **replicate** a subset of a relational ... In case of conflict, the central **database** is not ...

[Cited by 12](#) - [Related Articles](#) - [Web Search](#)

[The Bengal Database Replication System - group of 7 »](#)

T Ekenstam, C Matheny, P Reiher, GJ Popek - Distributed and Parallel Databases, 2001 - Springer

... updates by comparing update **timestamps** on replicated ... was made to "tune" the underlying

databases. ... storage overheads required to **replicate** particular data ...

[Cited by 15](#) - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

[Multiversion reconciliation for mobile databases - group of 10 »](#)

SH Phatak, BR Badrinath - Data Engineering, 1999. Proceedings., 15th International ..., 1999 - [ieeexplore.ieee.org](#)

... present a system architecture and a **replication** model for ... The **timestamps** reflect the order in which the ... which means it globally commits), the **database** state D ...

[Cited by 35](#) - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

[Taming aggressive replication in the Pangaea wide-area file system -](#)

10/661,412

[group of 18 »](#)

Y Saito, C Karamanolis, M Karlsson, M Mahalingam - ACM SIGOPS Operating Systems Review, 2002 - [portal.acm.org](#)

... they all use a centralized **database** to keep ... three challenges posed by optimistic **replication**: efficient and ... having each delta carry two **timestamps**: the old ...

[Cited by 82](#) - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

Transaction-Centric Reconciliation in Disconnected **Client-Server**

Databases - [group of 6 »](#)

SH Phatak, B Nath - Mobile Networks and Applications, 2004 - Springer

... In most systems the **replication** (hoarding) and local update ... The **timestamps** reflect

the order in which the ... it globally commits), the **database** state D "expands ...

[Cited by 11](#) - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

Sleepers and workaholics: caching strategies in mobile environments -

group of 3 »

D Barbará, T Imieliński - Proceedings of the 1994 ACM SIGMOD international conference ..., 1994 - [portal.acm.org](#)

... 3 The **replication** of the database between many ... asynchronously) in which only the

database items which have been ... the update's **timestamps**, for instance. ...

[Cited by 317](#) - [Related Articles](#) - [Web Search](#) - [BL Direct](#)

Archipelago: an island-based **file** system for highly available and scalable Internet services - [group of 9 »](#)

M Ji, E Felten, R Wang, JP Singh - USENIX Windows Systems Symposium, 2000 - [db.usenix.org](#)

... in addition to data distribution and metadata **replication**. ... in r in the ascending order of their **time stamps**. ... availability of an island-based **file** system with ...

[Cited by 30](#) - [Related Articles](#) - [Cached](#) - [Web Search](#)

Google 

Result Page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [Next](#)

[Google Home](#) - [About Google](#) - [About Google Scholar](#)

©2006 Google

Dialog DataStar[options](#)[logout](#)[feedback](#)[help](#)[databases](#)[easy search](#)**Advanced Search:****Inspec - 1898 to date (INZZ)**[limit](#)

Search history:

No.	Database	Search term	Info added since	Results	
CP		[Clipboard]		0	-
1	INZZ	replicat\$ WITH databases\$ WITH (file OR files OR folder OR folders) WITH timestamp\$	unrestricted	0	-
2	INZZ	replicat\$ WITH databases\$	unrestricted	363	show titles
3	INZZ	2 AND replicat\$ WITH (file OR files OR folder OR folders)	unrestricted	19	show titles
4	INZZ	3 AND timestamp\$	unrestricted	1	show titles

[hide](#) | [delete all search steps...](#) | [delete individual search steps...](#)Enter your search term(s): [Search tips](#) ☐ Thesaurus mapping Information added since: or:
(YYYYMMDD)[search](#)☐ Documents with images

Select special search terms from the following list(s):

- ☒ Publication year 1950-
- ☒ Publication year 1898-1949
- ☒ Inspec thesaurus - browse headings A-G
- ☒ Inspec thesaurus - browse headings H-Q
- ☒ Inspec thesaurus - browse headings R-Z
- ☒ Inspec thesaurus - enter a term
- ☒ Classification codes A: Physics, 0-1
- ☒ Classification codes A: Physics, 2-3

10/661,412

Dialog DataStar

options

logout

feedback

help

databases

search
page

titles

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save

locally as: PDF document

search strategy: do not include the search strategy

order

copy to
clipboard

Full text available at

USPTO Full Text Retrieval Options

open url

☒ document 1 of 1 [Order Document](#)

Inspec - 1898 to date (INZZ)

Accession number & update

0006190540 20051201.

Title

Handling of mutual conflicts in distributed **databases** using **timestamps**.

Source

Computer Journal, {Comput-J-UK}, 1998, vol. 41, no. 6, p. 376-85, 22 refs, CODEN: CMPJA6, ISSN: 0010-4620.

Publisher: Oxford University Press for British Comput. Soc, UK.

Author(s)

[Kumar-Madria-S.](#)

Author affiliation

Kumar Madria, S., Centre for Adv. Inf. Syst., Nanyang Technol. Inst., Singapore.

Abstract

In distributed **database** systems, it is desirable to allow read and write accesses to occur independently on **replicated** copies of **database files** in case of network partitions to increase availability. However, the system should detect mutual conflicts among the copies of the **database files** when sites from different partitions merge to form one partition. We present a **timestamp-based** algorithm for the detection of both write-write and read-write conflicts for a single **file** in distributed **databases** when sites from different partitions merge. Our algorithm allows read and write operations to occur in different network partitions simultaneously. When the sites from two different partitions merge, the algorithm detects and resolves both read-write and write-write conflicts with the help of stored **timestamps** using some additional information. Once the conflicts have been detected, we propose some reconciliation steps for the resolution of conflicts to bring the **file** into some consistent state. Our algorithm does not take into account the semantics of the transactions while detecting and resolving conflicts. Our algorithm will be useful in real-time systems where timeliness of operations is more important than response time (delayed commit).

Descriptors

CONCURRENCY-CONTROL; DATABASE-THEORY; DISTRIBUTED-DATABASES; REAL-TIME-SYSTEMS.

Classification codes

C6160B Distributed-databases*;

C4250 Database-theory.

Keywords

mutual-conflict-handling; **distributed-databases**; read-accesses; write-accesses; **replicated-database-file-copies**; network-partitions; availability; **timestamp-based-algorithm**; write-write-conflict-detection; read-write-conflict-detection; **stored-timestamps**; reconciliation-steps; real-time-systems; operation-timeliness.

Treatment codes

T Theoretical-or-mathematical.

Language

English.

Publication type

Journal-paper.

Availability

SICI: 0010-4620(1998)41:6L.376:HMCD; 1-O.

Publication year

1998.

Publication date

19980000.



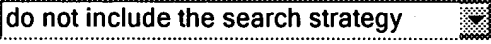


Edition

1999011.

Copyright statement

Copyright 1999 IEE.

(c) The Institution of Engineering & Technology

	locally as: 	search strategy: 
		

[Top](#) - [News & FAQs](#) - [Dialog](#)

© 2006 Dialog